INSTRUCTIONS MODBUS

Protocol Modbus.

The Modbus protocol is a communication standard for the industry that allows the connection of multiple devices to the network, where there is a master and multiple slaves. It allows an individual dialog master-slave and it also allows commands in broadcast format.

There are two types of strings in the Modbus Standard, the ASCII and the RTU (binary). In our case, for all our devices, we are using Modbus RTU.

In the RTU mode, the beginning and the end of the string are detected by means of silences of minimum 3,5 characters and it is used the CRC of 16 bits for the correction of errors. The length of the characters is prefixed to 8 bits; it allows parity odd or even or without parity, and 1 or 2 stop bits.

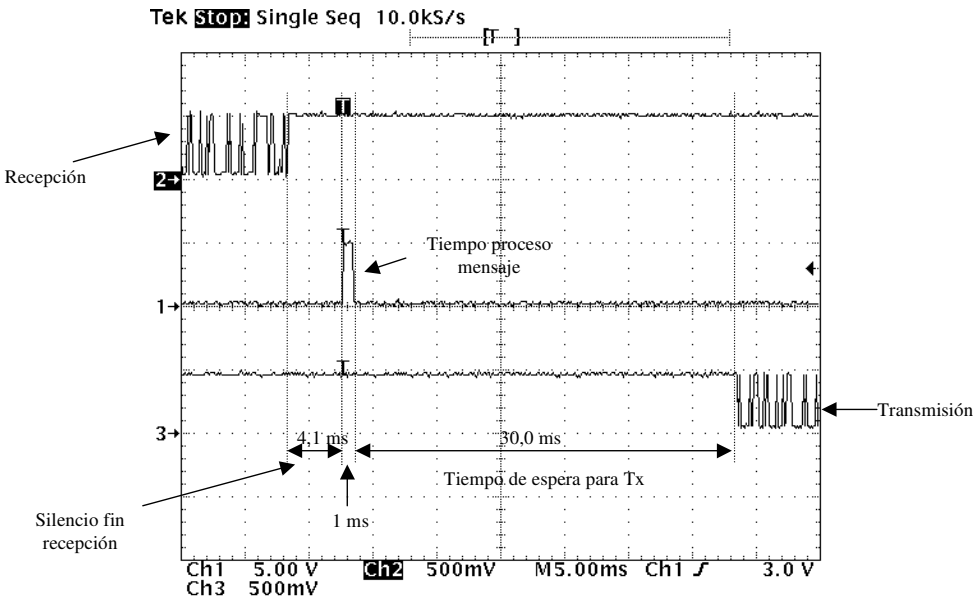
The typical message Modbus has the following format:

Beginning	Address	Function	Data	CRC	End
3,5 bytes	8 bits	8 bits	n x 8 bits	16 bits	3,5 bytes

Modbus accepts networks with devices which are working at different baud rates and as said the end of message is detected thanks to a silence of minimum 3.5 bytes. This implies that there can be devices in a network quicker or slower or which might leave longer or shorter silences. That is why there is usually a big silence between the detection of the end of message and the beginning of the answer from the slave. This is done to avoid conflicts between devices that may answer before all the rest of the devices have properly received the end of the message, as if it was not made like this, they would concatenate the message of the master with the replay of one of the slaves.

In the Series 3000 and 4000 there is left a time of approximately 30 ms, which is equivalent to 3.5 bytes at the slowest baud rate of 1200.

In the following graph, you can see the graph of a typical communication Modbus:



Implemented functions

Function 01 (01 Hex): Reading of the compacted relays.
Function 03 y 04 (03 y 04 Hex): Reading of integer registers.
Function 05 (05 Hex): Writing for a certain relay
Function 15 (0F Hex): Writing of multiple compacted relays.
Function 16 (10 Hex): Writing of multiple integer registers.

Variable description

They are basically split into 2 areas. The variables with margin of address, from 0 to 1000, are electrical variables usually checked by the user. The variables between the 1000 (03E8 Hex) and the 10000 (2710 Hex) are the ones which are related with the configuration of the device, which can be changed by the end user. We can also distinguish between two different types of variables, the relays and the integer registers.

VARIABLES RELAY TYPE:

Variable	Margin of address	Command Modbus Function	Type
Digital output port	0 to 7 (0000)	01, 05	R/W
Reset	2000 (07D0)	05	W

VARIABLES INTEGER TYPE:

Variable	Margin of address	Command Modbus Function	Type
Electrical variables	0 to 18, 120 to 138, 240 to 258	04	R
Config. Communications	1000 to 1002 (03E8)	04,16	R/W
Config. Setup 2	1100 to 1105 (044C)	04,16	R/W
Config. Alarm 1 (relay)	1150 to 1156 (047E)	04,16	R/W
Config. Alarm 2 (led)	1200 to 1206 (04B0)	04,16	R/W
Conf.Maximum demand[1]	1250 to 1251 (04E2)	04,16	R/W
Message starting screen	1300 to 1303 (0514)	04,16	R/W
Versión of the software	1400 to 1405 (0578)	04	R
Error code EEPROM	1500 (05DC)	04	R
Número de serie	10000 (2710)	04	R

[1] Command only for the KW3000 and KW4000

VARIABLES RELAY TYPE

Digital output port:
It allows to read and change the state of the digital output.
The outputs we are referring to are:

Address Modbus	Name output	Port micro	Answering bit
0000 (0000)	Relay 1	P6 bit 6	0
0001 (0001)	Relay 2 (led)	P6 bit 7	1

Command to read the state of the port:
Rx: NP0100000008*CRC*
Tx: NP0101XX*CRC*

XX is the byte of state of the relay. The most significant bit is always zero and does not correspond to any output.
CRC= Check-sum of 16 bits

To force the output from 0 to 1, it should be made individually with the command:
Tx: NP05AAAADD00*CRC*
Rx: NP05AAAADD00*CRC*

NP= Number of peripheral
AAAA= address of the relay.
DD= 00 if we want to force a zero, FF if we want to force a "1".

Reset:
Writing a "0" or "1" in this relay, the system makes a reset.
Command:
Tx: NP0507D0XX00*CRC*
Rx: Nothing.

XX can be 00 or FF. There is no answer, since immediately after receiving the command, the system is reset.

VARIABLES INTEGER TYPE

Modbus uses integer registers. However, the device uses different formats, for example for the peripheral number, the character is only 1 byte, while the PT ratio it is 4 bytes long. To simplify all the registers, all the vectors have been treated as gathered characters of 2 bytes and it is not allowed to write a separate part of these vectors. It is always mandatory to write the complete value, even though there might be an address for each pair of bytes. The only variable where it is allowed to ask for only a part of it is for the electrical variables, as there is not different formats in it.

Electrical variables:
It includes all the variables that the series 3000 and 4000 measure and calculate. The variables are in long format, so each one has two registers, since Modbus works with integer registers. Therefore, the address for reading a variable has to be always even. The number of registers has to be also even, as they have to be asked in pairs.
In the anex1 of this manual, you will find the list of variables with their corresponding Modbus address. The variables are split in three areas, the one of instantaneous values, the ones between the address 0 to the 18. The second area is the one for the maximums, the ones between the address 120 to the 138 and the third area is the one for the minimums which is located between the address 240 to the 248.

Reading command of electrical variables:
Tx: NP04AAAANNNN*CRC*
Rx: NP04BBDDDD... DDDD*CRC*

NP= Peripheral number
AAAA= address of the first register you are asking for
NNNN= Number of registers you are asking for.
BB= Number of sent bytes.
DD= 1 byte of data of the asked variables. Each 4 bytes is the value of a variable.

The address and the number of registers have to be always an even value, as every variable is made out of two integer registers (1 long).

Communication configuration:
The variables gathered between the location 1000 (3E8 Hex) and the 1002 (3EA Hex) are modifying the configuration of the serial port of the device. The same operation can be made through the Set up 1.
The configuration of the communications is stored in bytes, so in every Modbus register, we will have two variables of communications. There has to be read and written three registers at a time, they cannot be requested separately.

The registers are set in the following way:

Modbus address	Variable that modifies	Valid Margin for the data
1000H	Protocol	0- Modbus
1000L	Number of peripheral	1 to 255 (1 default)
1001H	Baud rate (baud)	0- 1200, 1- 2400, 2- 4800, 3- 9600 (def), 4- 19200
1001L	Parity	0- No (def) 1- odd 2- even
1002H	Length of the data bits	1- 8 bits
1002L	Stop bits	0- 1 bit (def), 1- 2 bits.

Writting command:
Tx: NP1003E8000306DDDDDDDDDDDD*CRC*
Rx: NP1003E80003*CRC*

Reading command:
Tx: NP0403E80003*CRC*
Rx: NP0406DDDDDDDDDDDD*CRC*

If an incorrect parameter change is made and stored in the meter, with a wrong parameter, the device will be configured with the default setting after reset.

Setup Configuration:
These variables allow you to read and modify the programmable parameters of the device, except the alarms, which are treated separately.
They go from the address 1100 (44C Hex) to the 1105 (451 Hex).
The device does not check that the sent parameters are within the acceptable margins. In the attached table, you can see the registers:

Modbus Address	Variable that modifies	Margin for the data
1100,1101	Voltage Primary	0 to 99999 (1 def)
1102	Voltage Secondary	0 to 999 (1 def)
1103	Current Primary	0 to 10000 (1 or 5 default.)
1104H	Reserved	0
1104L	Reserved	0
1105H	Reserved	0
1105L	Disconnecting time backlight	0 to 99 segundos

As mentioned previously, all the parameters have to be configured at the same time, due to the wide variety of formats that there are.
The starting point of these variables is the 1100 (044C).
Writting Comand:
Tx: NP10044C00060CDDDDDDDDDDDDDDDDDDDDDDDD*CRC*
Rx: NP10044C0006*CRC*

Reading command:
Tx: NP04044C0006*CRC*
Rx: NP040CDDDDDDDDDDDDDDDDDDDDDDDD*CRC*

Configuration Alarm 1:
These variables are programming the alarm 1 as in the set up 2.
They are going from the address 1150 (47E Hex) to the 1155 (483 Hex).

The parameters are:

Modbus address	Variable that modifies	Margin of the data
1150, 1151	Value MAX or energy pulse	Depending on the variable (0 default)
1152, 1153	Value MIN	Depending on the variable (0 default)
1154	Delay	0 to 9999 (0 default)
1155H	Number of variable	0 to 10 (0 default)
1155L	Free	Any value

Writting command:
Tx: NP10047E00060CDDDDDDDDDDDDDDDDDDDDDDXX*CRC*
Rx: NP10047E0006*CRC*

Reading command:
Tx: NP04047E0006*CRC*

Rx: NP040CDDDDDDDDDDDDDDDDDDDDXX*CRC*

Configuration Alarm 2:
These variables are exactly the same ones as for the alarm 1, it is only the Modbus address that changes and that in the alarm 2 goes from 1200 (4B0 Hex) to 1205 (4B5 Hex).

The parameters are:

Modbus adress	Variable that modifies	Margin of the data
1200, 1201	Value MAX or energy pulse	Depending on the variable (0 default)
1202, 1203	Value MIN	Depending on the variable (0 default)
1204	Delay	0 to 9999 (0 defecto)
1205H	Number of variable	0 to 10 (0 default)
1205L	Free	Any value

Writting command:
Tx: NP1004B000060CDDDDDDDDDDDDDDDDDDDDXX*CRC*
Rx: NP1004B00006*CRC*

Reading command:
Tx: NP0404B00006*CRC*
Rx: NP040CDDDDDDDDDDDDDDDDDDDD00*CRC*

Configuration maximum demand:
This command is only for the 3000+ and 4000+

Modbus Address	Variable that modifies	Margin of the data
1250	Variable to calculate Pd	0 (no Pd), 16 (KwIII)
1251	Period	1 - 60 minutes

Writting Comand:
Tx: NP1004E2000204DDDDDDDD*CRC*
Rx: NP1004E20002*CRC*

Reading Comand:
Tx: NP0404E20002*CRC*
Rx: NP0404DDDDDDDD*CRC*

Version of the software:
This variable indicates the version of the software loaded on the device.
The version number is a string of 6 characters ASCII, that correspond to 3 Modbus registers.
The last byte will be always 0 (end of string in C).

Reading Command:
Tx: NP0405780003*CRC*
Rx: NP0406DDDDDDDDDDDD*CRC*

For example, if the answer is 040620342E303100, the version would be:
20 → Space; 34 → 4; 2E → '.' (point); 30 → 0 ; 31 → 1

Serial number:
These variables Modbus allow you to read the serial number of the device. They include two registers, as the serial number is a long. Therefore, the address is the 10000 (2710 Hex) and 10001 (2713 Hex). Both registers have to be asked at the same time.

Reading Command:
Tx: NP0427100002*CRC*
Rx: NP0404DDDDDDDD*CRC*

ANEX 1: 3-PHASE ELECTRICAL VARIABLES

			MODBUS VARIABLES			
VARIABLE	**SYMBOL**	**CODE**	**INSTANT ANEOUS**	**MÁXIMUM**	**MÍNIMUM**	**UNITS**
Voltaje phase 1	V 1	1	01-02	120-121	240-241	V x10
Current phase 1	A 1	2	03-04	122-123	242-243	mA
Active power phase 1	Kw 1	3	05-06	124-125	244-245	W
Voltaje phase 2	V 2	4	07-08	126-127	246-247	V x10
Current phase 2	A 2	5	09-10	128-129	248-249	mA
Active power phase 2	Kw 2	6	11-12	130-131	250-251	W
Voltaje phase 3	V 3	7	13-14	132-133	252-253	V x10
Corrent phase 3	A 3	8	15-16	134-135	254-255	mA
Active power phase 3	Kw 3	9	17-18	136-137	256-257	W
Active energy	Kwh III	10	19-20	138-139	258-259	Wh
Máx. Maximum demand	Md	11	21-22			W